

Communication Protocols for Embedded Systems

There's more to connecting multiple CPUs than just stringing wires or cable. Your choice of network protocol, in particular, will determine system performance.

The past few years have seen a growing trend to dramatically increase the embedded electronics content of automobiles, elevators, building climate control systems, jet aircraft engines, and other traditionally electro-mechanically controlled systems. In many large systems, this increasing electronics content is accompanied by a proliferation of sub-systems with separate CPUs.

The increase in the number of processors in a system is often driven by computation and I/O growth. In some development environments, the increase may also be driven by a need to ease system integration burdens among multiple design groups or to provide system flexibility through "smart sensors" and "smart actuators." Whatever the reasons, once there is more than one CPU in a system, there must be some means of communication to coordinate action.

While some high-end embedded systems communicate over a VME backplane or similar arrangement, the embedded systems we're working on use physically distributed CPUs involving some sort of local area network (LAN), also called a multiplexed network or a communication bus. At the heart of the LAN is the media access protocol, which picks the next

transmitter for access to the shared network medium, typically a wire, fiber, or RF frequency.

In this article, we will discuss the special considerations for networking real-time embedded systems, and look at several media access protocols that demonstrate fundamentally different ways of accessing the shared medium. The protocols are: connection-oriented protocols, polling, time division multiple access (TDMA), token ring, token bus, binary countdown, carrier sense multiple access with collision detection (CSMA/CD), and carrier sense multiple access with collision avoidance (CSMA/CA). For each of these, we will evaluate the strengths and weaknesses against special considerations. A protocol tradeoff chart will enable you to select a protocol to fit your needs. While no protocol is perfect for all purposes, a variation of CSMA/CA offers the most versatility for many embedded systems.¹

SPECIAL CONSIDERATIONS

In practice, we have found that embedded real-time networks require high efficiency, deterministic latency, operational robustness, configuration flexibility, and low cost per node.

Because cost limits the network bandwidth available to many applica-

Nance Paternoster

tions, protocol efficiency (message bits delivered compared to raw network bandwidth) is very important. The embedded systems we have studied are characterized by a predominance of short, periodic messages. So, an obvious optimization is to reduce overhead bits used for message packaging and routing. (It is not unusual for eight bits of data to be packed in a message that is 32 or even 64 bits long.)

Once message overhead has been reduced as much as possible, media access overhead must be reduced. For the most part, this is accomplished by minimizing the network bandwidth consumed by arbitration (such as passing a token or resolving collision conflict). Because worst-case behavior is typically important, efficiency should be evaluated for both light and heavy traffic. For example, CSMA/CD (often used in workstation LANs) is highly efficient for light traffic but gives poor performance if heavily loaded, while token bus protocols have the reverse properties.

Determinacy, or the ability to calculate worst-case response time, is important for meeting the real-time constraints of many embedded control applications. A prioritization capability is usually included in systems to improve determinacy of messages for time-critical tasks such as exception handling and high-speed loop control. Priorities can be assigned by node number or message type. Additionally, protocols can support local or global priority mechanisms. In local prioritization, each node gets a turn at the network in sequence and sends its highest priority queued message (thus potentially forcing a very high-priority message to wait for other nodes to have their turns first). In global prioritization, the highest priority message in the entire system is always transmitted first. This mechanism, which is fundamentally enabled by the media access protocol, is highly desirable for many safety-critical applications.

Many applications require robust

Because worst-case behavior is typically important, efficiency should be evaluated for both light and heavy traffic.

operation under extreme conditions. We call a protocol robust if it can quickly detect and recover from errors (duplicate or lost tokens, for example), added nodes, and deleted nodes. In some systems, it's important to quickly recover from a reset or power glitch that forces a restart of the network.

Varied operating environments may dictate use of a media access protocol that is flexible in supporting multiple media and mixed topologies. Portions of a system may require expensive fiber in noisy environments, while other portions can tolerate low-cost twisted pair wires in benign environments. A bus topology may be optimum for wires, but a ring or star topology may be needed for fiber.

A vital consideration is the cost per node. In this article, the order of the media access discussion progresses from very simple to complex, high-performance protocols. Simple protocols require less hardware and software resources and are therefore likely to be less expensive. For extremely cost-sensitive high-volume applications, these protocols are good candidates. However, for growing applications, more advanced protocols provide a stronger foundation. In general, costs are decreasing over time due to



Communication Protocols

advances in IC manufacturing technology and the increasing availability of off-the-shelf protocols. Consequently, we envision advanced cost-effective protocols used in many embedded applications.

MEDIA ACCESS PROTOCOLS

Now that we have a feel for the issues in embedded networks, let's examine various commonly available media access protocols. While many variations and combinations are possible, we'll discuss the plain versions of each protocol.

Before LANs became popular, connection-oriented protocols were heavily used to connect remote terminals to mainframes. These protocols support only two nodes per physical transmission medium and are typically connected via modem with serial lines. Figure 1 shows an example of a four-processor network using this protocol. Communication between nodes not physically connected requires multiple transmissions through intermediate nodes. These protocols are deterministic between directly connected nodes. For indirectly connected nodes, latency can be high.

For an embedded system with modest communication requirements, this might be a cost-effective protocol (readily available hardware and software from mature technology). For demanding applications, nodes that handle a lot of pass-through traffic can become swamped, prohibiting use of low-cost nodes in a large system. Sometimes, this type of protocol is combined with a more complex communication system to provide backward compatibility to older systems or to allow simple remote modem access to the system (such as BACnet). This type of protocol is used by the X.25 public network standard (network services offered by telephone companies) and IBM's system network architecture (SNA).²

Polling is one of the more popular protocols for embedded systems because of its simplicity and determi-

nacy. In this protocol, a centrally assigned master periodically sends a polling message to the slave nodes, giving them explicit permission to transmit on the network.

Figure 2 shows the polling order (dotted lines) of a simple four-node bus network. The majority of the protocol software is stored in the master and the communication work of slave nodes is minimal (therefore, the network costs tend to be smaller). This protocol is ideal for a centralized data-acquisition system where peer-to-peer communication and global prioritization are not required. However, the single-point-of-failure from the master node (or the cost of installing redundant master hardware) is often unacceptable. Additionally, the polling process consumes considerable bandwidth regardless of network load (poor

efficiency). These protocols have been standardized by the military (MIL-STD-1553B) for aircraft subsystem communications. Some variants of this protocol allow inter-slave communication through the master as well as improved robustness by using multiple masters (as does Profibus).

TIME DIVISION MULTIPLE ACCESS

Time division multiple access (TDMA) is heavily used in satellite communications but is applicable to LANs as well.³ In this protocol, a network master broadcasts a frame sync signal before each round of messages to synchronize clocks of all the nodes. After the sync, each node transmits during its uniquely allocated time slice, as shown in Figure 3. Performance is similar to polling, but with greater efficiency at heavy loads

FIGURE 1

An example network using connection-oriented protocols.

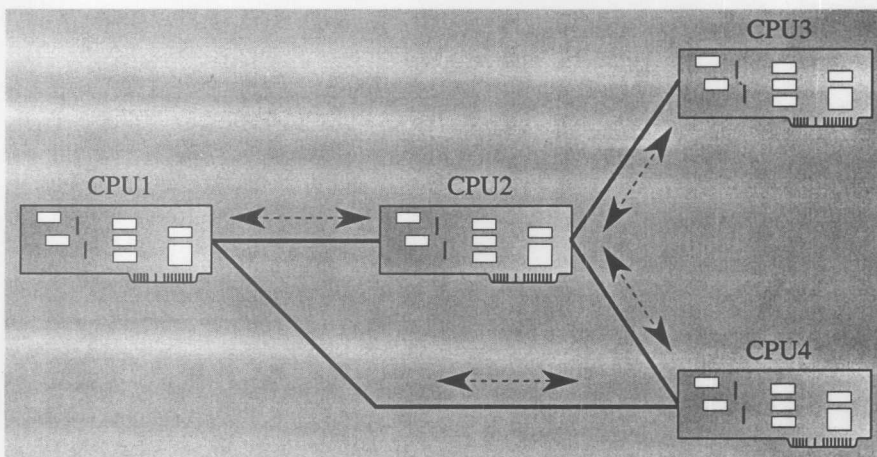
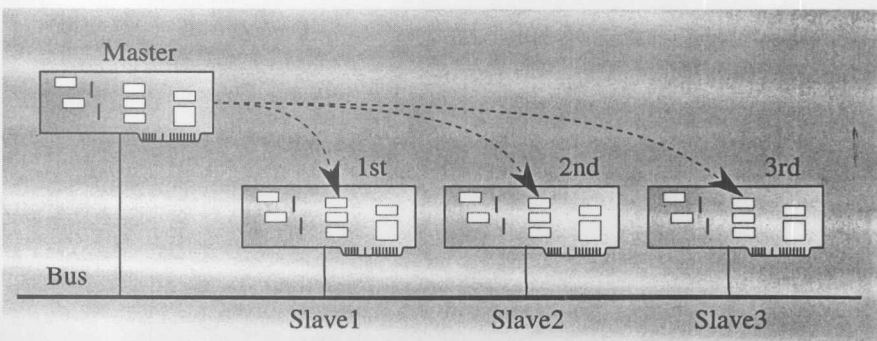


FIGURE 2

Master node sequentially polling slave nodes for information..



Communication Protocols

due to elimination of individual polling messages. Costs for slave nodes are greater with TDMA than with polling, because each slave node must have a stable time base to measure slices. An additional weakness for TDMA is the need for fixed-length messages to fit into time slices. In some TDMA variations, unused slices are truncated by tacit agreement among nodes. Time-based protocols have been popular in aerospace applications. For example, digital autonomous terminal access communications (DATAC) is being used by NASA and Boeing.

In a token ring network, the nodes are connected in a ring-like structure using point-to-point links as shown in Figure 4. A special token signal is passed from node to node around the ring. When a node has something to send, it stops the token circulation, sends its message all the way around the ring, and passes the token on. Since worst-case token waiting time can easily be calculated, this protocol is deterministic. Under light traffic, token ring has moderate token passing overhead. However, the protocol provides efficient throughput under heavy traffic conditions since idle token passing is minimized.

A frequent implementation strategy is to have a one-bit delay at each node, so a token can visit all nodes in $N+T$ bit times, where N is the number of nodes and T is the number of bits in the token. Global prioritization is accomplished by altering the priority field of the token as it visits the nodes. This field enables only the nodes with a high priority to send messages on the network. Initialization of the token message and detection of accidentally duplicated or lost tokens adds complexity and cost to the protocol.

A break in the cable or a failed node disabling the entire network is a common concern for many users. Consequently, node bypass hardware and dual rings are used to address this concern at additional cost. Because the ring connections themselves are point-to-point, it is well suited for fiber

optics. So, many LANs and wide area networks (WANs) are moving to this type of protocol. For example, fiber distributed data interface (FDDI) uses dual counter-rotating rings to achieve higher reliability than bus or star topologies.

TOKEN BUS

The operation of a token bus is very similar to a token ring—a token is passed from node to node in a virtual ring as in Figure 5. The holder of the token has the access to the network. Like token ring, token bus works well under heavy traffic with a high degree of determinacy.

However, token bus broadcasts the message simultaneously to all nodes instead of passing it bit-by-bit along a physical ring. The minimum time for a token to traverse the logical ring of nodes is thus $N \cdot T$ bit times instead of $N+T$ bit times as in token ring (because there is no parallelism in the connections). This makes global prioritization of messages largely impractical.

Unlike unidirectional token ring, a break in the cable or a failed node does not necessarily disable the entire network. A lengthy reconfiguration process, where each node identifies its neighbors, is used to maintain the virtual ring when nodes are added or

FIGURE 3

The time slices of TDMA protocols.

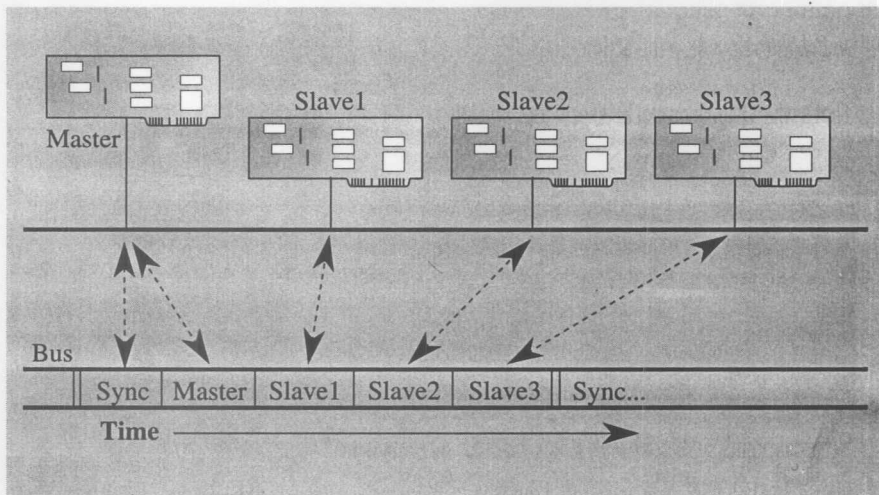
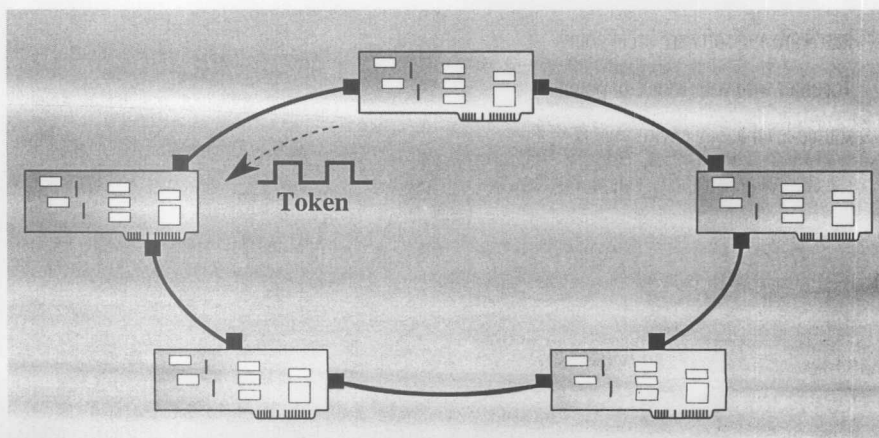


FIGURE 4

Token passing in the token ring networks.



Communication Protocols

deleted from the network. Because bus-like topologies are well suited for manufacturing plants, manufacturing automation protocol (MAP) adopted this protocol. Additionally, attached resource computer network (ARCnet)⁴ uses this protocol for LAN connectivity and process control. Adaptive Networks' PLC-192 power line carrier chip uses a hybrid token bus protocol: under light traffic, nodes dynamically join and leave from the logical ring—under heavy traffic, all nodes join the ring to maintain stability.

In binary countdown, also known as the bit dominance protocol, all nodes wait for an idle channel before transmitting a message. Competing nodes (transmitting simultaneously) resolve contention by broadcasting a signal based on their unique node identification value. The transmission medium must have the characteristic that one value (say, a "1") overrides the opposite value (a "0"). During this transmission, a node drops out of the competition if it detects a dominant signal opposite to its own, as shown in Figure 6. Thus, if a "1" signal is dominant, the highest numbered transmitting node wins the competition and gains ownership of the channel.

Global prioritization can be achieved by arbitrating over message ID values rather than the node IDs. Since the arbitration is part of the message, this protocol has good throughput and high efficiency. Additionally, the protocol is more robust because node configuration (transmission order) is not required, and inactive nodes are ignored. However, since all messages are prioritized, there is no simple way to guarantee equally fair access among all nodes under heavily loaded conditions. Also, some transmission techniques (such as current-mode transformer coupling commonly used in high-noise environments) aren't compatible with the bit dominance requirement. Using this protocol, Bosch developed the controller area network (CAN) specification for automotive applications.⁵ The Society of

Automotive Engineers standard SAE J-1850 also uses this protocol.

CARRIER SENSE MULTIPLE ACCESS

Carrier sense multiple access with collision detection (CSMA/CD) has been widely researched with a large number of published variations. In the simplest case, a node waits for the network to go idle before transmission (as in binary countdown). If multiple stations transmit almost simultaneously (within a round-trip transmission delay on the network), the messages collide, as in Figure 7. The nodes must detect this collision and resolve it by waiting for a random time before retrying.

The key advantage to this protocol is that, in principle, it supports an unlimited number of nodes that don't require preallocated slots or inclusion in token

passing activities. CSMA/CD allows nodes to enter and leave the network without requiring network initialization and configuration. For light traffic conditions, overhead is very small. However, under heavy traffic, the overhead is unbounded due to high probability of repeated collisions. Consequently, this protocol has poor determinacy and low efficiency. Furthermore, detecting collisions may require analog circuitry, adding to the system expense. In fact, if the network environment is very noisy or the wiring runs are long and of poor quality, collision detection may not work at all. The popular Ethernet protocol used in workstation LANs is based on this protocol.

Many hybrid protocols combine the light traffic efficiency of CSMA/CD with the heavy traffic efficiency of

FIGURE 5

Token passing in token bus protocols.

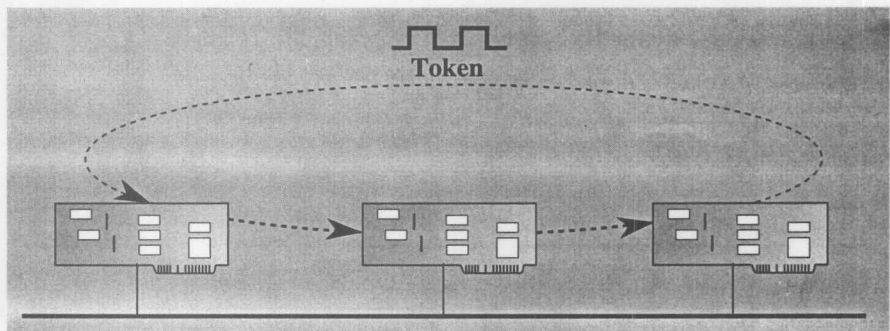
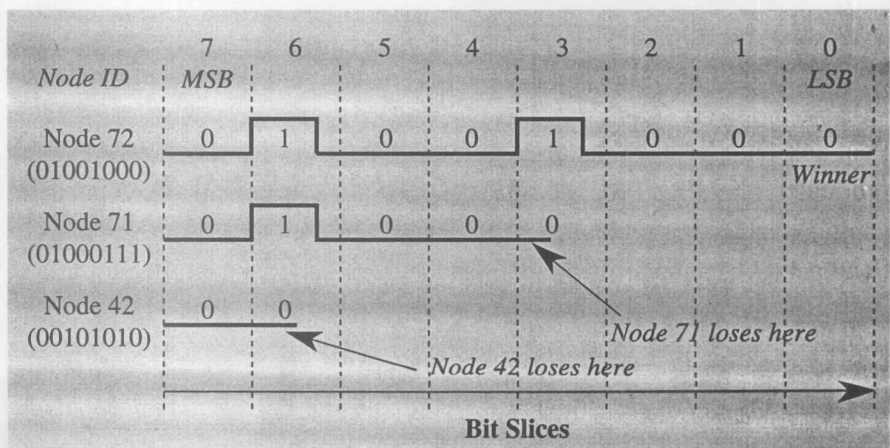


FIGURE 6

Arbitration in binary countdown protocols.



Communication Protocols

token-based protocols. The resulting protocols are often called carrier sense multiple access with collision avoidance (CSMA/CA). As in CSMA/CD, nodes transmit after detecting an idle channel. However, if two or more stations collide, a jam signal is sent on the network to notify all nodes of collision, synchronize clocks, and start contention time slots. Each contention time slot, typically just over a network round-trip propagation delay time, is assigned to a particular station. Each station is allowed to initiate transmission during its contention slot. Figure 8 shows a slot progression for a three node network. In this example, transmitters 2 and 3 collide and initiate a jam. Contention slots follow the jam signal. Since processor 1 has nothing to send, slot1 goes idle. Transmitter 2 starts sending its message during slot2. Other stations detect the message, and stop the slot progression.

After the end of the message, all nodes initiate new contention slots. However, to ensure fairness and determinacy, the slots are rotated (change positions) after each transmission. Additionally, the priority slots (pslots) can precede each slot progression to support global prioritization for high-priority messages. The network returns to an idle state when all the slots go unused.

The contention slots in CSMA/CA protocol help avoid collisions. In general, there are two distinct variations of CSMA/CA protocols. If the number of slots equals the number of stations, the protocol is called reservation CSMA (RCSMA). The RCSMA variation works efficiently under all traffic conditions.⁶ However, because of the one-to-one relation of the node to the slot, RCSMA is not practical for a network with a large number of nodes. In another variation, the number of slots is less than the number of stations, and the slot assignments are randomly allocated to minimize collisions. Echelon's local operating network (LON) uses the latter variation and dynamically varies the number of slots based on

FIGURE 7

Collisions in CSMA/CD networks.

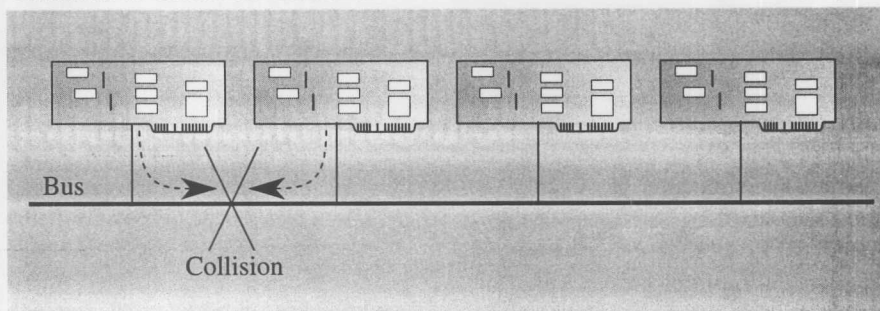
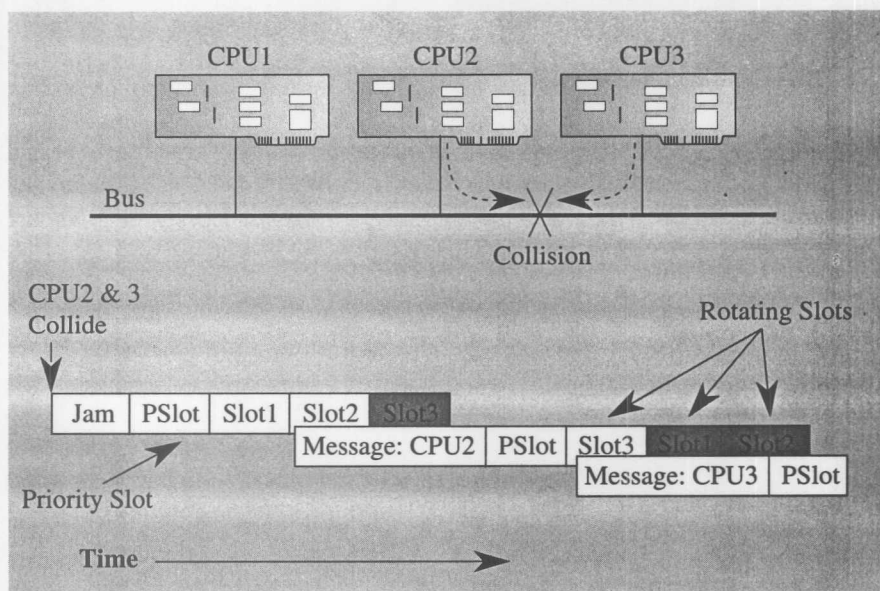


FIGURE 8

Slot progression in CSMA/CA protocols.



expected traffic prediction.⁷ Unlike CSMA/CD, there are ways to eliminate the need for collision detection hardware, such as sending dummy messages that keep slots going in the absence of network traffic.

PROTOCOL TRADEOFFS

We have described the major media access protocols and noted clear differences. Table 1 summarizes some of the common traits and our assessment of their strengths and weaknesses for embedded real-time applications. The important points to take into consideration when evaluating alternatives are:

- Polling, TDMA, and connection-

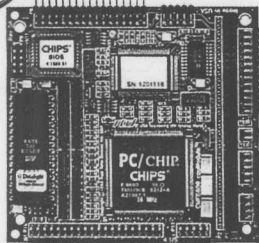
based protocols are simple, but may not provide sufficient flexibility for advanced systems.

- Token-based protocols are predictable, but can have high overhead and require complex software to maintain robustness.
- Binary count-down protocols rely heavily on the bit dominance characteristics of the physical medium.
- CSMA/CD is a poor choice for hard real-time systems with heavy traffic.

For our embedded systems, we have found that CSMA/CA, particularly RCSMA, is a good choice. While your application will no doubt have characteristics that are somewhat different

Replace Four Conventional PC/104 Modules with One SuperXT™ CMF8680 cpuModule™

Embedded PC/XT Controller with
Intelligent Power Management



PC/104 Compliant
Actual Size: 3.6 x 3.8 x 0.6"

\$449
100 pcs.

- PC/XT compatibility with 286 emulation
- 14 MHz, 16-bit 8086 CPU
- +5V only; 1.6W at 14.3 MHz, 1W at 7.2 MHz
- Intelligent sleep modes, 0.1W in Suspend
- ROM-DOS and RTD enhanced BIOS
- Compatible with MS-DOS & real-time operating systems
- 1M bootable Solid State Disk & free software
- 4K-bit configuration EEPROM (2K for user)
- 2M on-board DRAM
- IDE & floppy interfaces
- CGA CRT/LCD controller
- Two RS-232 ports, one RS-485 port
- Parallel, XT keyboard & speaker ports
- Optional X-Y keypad scanning/PCMCIA interface
- Watchdog timer & real-time clock

Expand This Or Any PC/104 System
with the

CM106 Super VGA Controller utilityModule™

- Mono/color STN & TFT flat panel support
- Simultaneous CRT & LCD operation
- Resolution to 1024 x 768 pixels
- Displays up to 256 colors

\$223
100 pcs.

Speed Product Development with the DS8680 Development System

Your DS8680 includes the CMF8680, CM102 keypad scanning/PCMCIA, CM104 with 1.8" 85MB hard drive, CM106 SVGA controller & DM5406 12-bit, 100 kHz dataModule™ in an enclosure with external power supply, 3.5" floppy, keyboard, keypad, TB50 terminal board, SIGNAL*VIEW™, SIGNAL*MATH™, MS-DOS, SSD software & rtdLinux™ for just

\$2950.

For more information on our PC/104 and
ISA bus products, call today.



Real Time Devices USA

200 Innovation Blvd. • P.O. Box 906
State College, PA 16803 USA
(814) 234-8087 / Fax: (814) 234-5218

RTD Europa • RTD Scandinavia

Real Time Devices is a founder of the PC/104 Consortium.

TABLE 1
Media access tradeoffs.

| ↑ Good — Ok ↓ Poor | Efficiency Light Traffic | Efficiency Heavy Traffic | Deter- minacy | Priori- tization | Robust- ness | Physical Layer Flexibil. | Low Cost/ Node |
|--------------------------|--------------------------------|--------------------------------|------------------|---------------------|-----------------|--------------------------------|----------------------|
| Connection | — | ↓ | ↑ | — | ↑ | ↓ | — |
| Polling | ↓ | — | ↑ | ↓ | ↓ | ↑ | — |
| TDMA | ↓ | ↑ | ↑ | ↓ | ↓ | ↑ | ↓ |
| Token Ring | ↑ | ↑ | ↑ | ↑ | — | — | — |
| Token Bus | — | ↑ | ↑ | — | — | ↑ | — |
| Binary Cnt. | ↑ | ↑ | — | — | ↑ | ↓ | ↑ |
| CSMA/CD | ↑ | ↓ | ↓ | — | ↑ | ↑ | — |
| CSMA/CA | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ |

than ours, this article's discussion of the special considerations and media access protocol strengths and weaknesses should allow you to select the best protocol to match your needs. We believe the electronic contents of embedded systems will continue to grow, and communication networks provide strong foundation for supporting this growth. **ESP**

Bhargav Upender is an associate research engineer at United Technologies Research Center. Currently, he is exploring novel architectures and supporting protocols for distributed embedded systems. He holds a BS in electrical engineering from the University of Connecticut and an MS in electrical engineering from Cornell University in New York. He can be contacted electronically at barg@utrc.utrc.com.

Philip Koopman is a principal research engineer at United Technologies Research Center. He currently designs and evaluates architectures and communication protocols for a variety of embedded applications. He has previously worked as an embedded CPU architect and a Navy submarine officer. Koopman holds a BS and MS in computer engineering from Rensselaer Polytechnic Institute

and a PhD in computer engineering from Carnegie Mellon University. He may be reached electronically at koopman@utrc.utrc.com.

REFERENCES

1. Upender, B. P. and Koopman, P. J., *Embedded Communication Protocol Options*, San Jose, CA: Proceedings of the Embedded Systems Conference, October 1993.
2. Tanenbaum, A. S., *Computer Networks*, 2nd ed., Englewood Cliffs, NJ: Prentice-Hall, 1989.
3. Stallings, W. S., *Data and Computer Communications*, 3rd ed., New York: Macmillan, 1991.
4. Hoswell, Katherine S. and Thomas, George M., *ARCnet Factory LAN Primer*, Second Printing, Downers Grove, Illinois: Contemporary Control Systems Inc., 1988.
5. Bosch, CAN Specification, ver. 2.0, Robert Bosch GmbH, Stuttgart: 1991.
6. Chen and Li, *Reservation CSMA/CD: A Multiple Access Protocol for LANs*, IEEE Journal on Selected Areas in Communications, February 1989.
7. *Enhanced Media Access Control with Echelon's LonTalk Protocol*, LonWorks Engineering Bulletin, Echelon Corp., August 1991.